

BTP : Shepherding of multievader system with optimal pursuer trajectory

Saqib Azim - 150070031

Advisor : Prof. Debraj Chakraborty

Electrical Engineering Department
Indian Institute of Technology, Bombay

Thursday 9th May, 2019

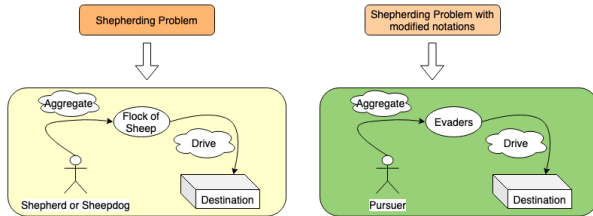
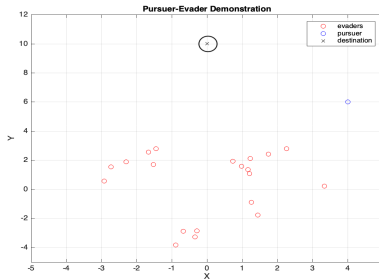


Figure: Overview of shepherding problem



General Problem Description

Assumptions

- N_e evaders and N_p pursuers located in a 2-dim-space (can be extended to N-dim-space)
- Each agent (pursuer as well as evader) has knowledge about the state of all other agents

Task of the Pursuer

- Drive the evaders to an ϵ radius of a predefined fixed destination point z based on some interaction rules between the pursuer and evaders
- Try to minimize the trajectory length of the pursuer

Interaction Rules

- 1 Repulsion force acting on evaders due to the pursuer $= F_r$
- 2 Attractive force on an evader towards the centroid of its n -nearest evaders $= F_a$
- 3 Repulsion acting between two evaders when $\|d(e_i, e_j, t)\|_2 \leq r_a$ (some threshold) $= f_r$
- 4 Thus, total force acting on an evader $F = F_r + F_a + f_r$

Below are some common notations which will be used throughout this presentation

- N_e denotes the number of evaders (≥ 1)
- N_p denotes the number of pursuers ($=1$)
- Let $p(t) \in R^2$ denote the position of the pursuer at time instant t
- Let $e_i(t) \in R^2$ denote the position of the i^{th} evader at time instant t
- Let $z(t) = z \in R^2$ denote the predefined fixed destination point
- $d(e_i, p, t) = e_i(t) - p(t)$ is the line of sight vector pointing from the pursuer towards the i^{th} evader at time instant t
- v_{emax} denotes maximum velocity of an evader
- v_{pmax} denotes maximum velocity of the pursuer
- v_{pmin} denotes minimum velocity of the pursuer
- T denotes the time when the objective has been achieved

Objective Function

$$\underset{p(t)}{\text{minimize}} \int_0^T \sqrt{\dot{p}(t)^T \dot{p}(t)} dt$$

Constraints

$$\|e_i(T) - z\|_2 \leq \epsilon, \quad i = 1, \dots, N_e \quad (1)$$

$$v_{pmin} \leq \|\dot{p}(t)\|_2 \leq v_{pmax}, \quad 0 < t < T \quad (2)$$

$\dot{e}_i(t)$ = velocity component due to repulsion from pursuer ($v_{repl}(t)$) +
velocity component due to attraction towards centroid ($v_{attr}(t)$)

$$v_{repl}(t) = v_{emax_repl} \exp(-k_1 \|d(e_i, p, t)\|_2) \frac{(1 + \cos\theta_{i,t})}{2} \frac{d(e_i, p, t)}{\|d(e_i, p, t)\|_2}, \quad i = 1, \dots, N_e, 0 < t < T \quad (3)$$

$\cos\theta_{i,t} = \frac{\dot{p}(t) \cdot d(e_i, p, t)}{\|\dot{p}(t)\|_2 \|d(e_i, p, t)\|_2}$, which represents the angle between pursuer velocity vector $\dot{p}(t)$ and pursuer-evader line of sight vector $d(e_i, p, t)$

Constraints (continued)

$$v_{attr}(t) = v_{max_attr} \exp(-k_1 \|d(e_i, p, t)\|_2 + k_2 \|d(c_{i,n}, e_i, t)\|_2) \frac{d(c_{i,n}, e_i, t)}{\|d(c_{i,n}, e_i, t)\|_2} \quad (4)$$

$c_{i,n}$: centroid of n-nearest neighbour of i^{th} evader

$$k_1, k_2 \geq 0$$

- Conceptually, $\|v_{attr}(t)\|_2 \propto \frac{1}{\|d(e_i, p, t)\|_2}$. Also, $\|v_{attr}(t)\|_2 \propto \|d(c_{i,n}, e_i, t)\|_2$
- $\|v_{attr}\|_2$ can be exponentially exploding. To make it stable, take $k_2 \ll k_1$ or $k_2 = 0$
- When N_e is large and $F = F_r$, driving the evaders to the destination is difficult
- But when N_e is small and $F = F_r$, only the repulsive force on evaders by pursuer may suffice to drive them towards destination

Objective Function

$$\underset{p(t)}{\text{minimize}} \int_0^T \sqrt{\dot{p}(t)^T \dot{p}(t)} dt \quad (5)$$

Constraints

$$\|e_i(T) - z\|_2 \leq \epsilon, \quad i = 1, \dots, N_e \quad (6)$$

$$v_{pmin} \leq \|\dot{p}(t)\|_2 \leq v_{pmax}, \quad 0 < t < T \quad (7)$$

$$\dot{e}_i(t) = v_{e_{repl}}(t) = v_{e_{max_repl}} \exp(-k_1 \|d(e_i, p, t)\|_2) \frac{(1 + \cos\theta_{i,t})}{2} \frac{d(e_i, p, t)}{\|d(e_i, p, t)\|} \quad (8)$$

$$\cos\theta_{i,t} = \frac{\dot{p}(t) \cdot d(e_i, p, t)}{\|\dot{p}(t)\| \|d(e_i, p, t)\|}$$

Algorithm

Keeping the pursuer speed to be fixed : $\|\dot{p}(t)\|_2 = v_p(t) = c$ (constant)

At each time step t , the pursuer moves in that direction which minimizes the below cost function

$$J(t) = \alpha[\|d(e_1, z, t+1)\|_2 + \|d(e_2, z, t+1)\|_2] + (1 - \alpha)\|d(e_1, e_2, t+1)\|_2 \quad (9)$$

$$\underset{\hat{v}_p(t)}{\text{minimize}} \quad J(t)$$

Repeat until $(\|d(e_1, z, t)\|_2 \leq \epsilon \text{ and } \|d(e_2, z, t)\|_2 \leq \epsilon)$

- Estimate the direction $\hat{\theta}$ in which the pursuer should move such that the cost function is minimized
- Move the pursuer in the direction $\hat{\theta}$

Results

- Cost function decreases and then saturates at some finite non-zero value
- This approach requires a lot of tweaking of parameters such as α , constant speed (c), etc. on case-to-case basis

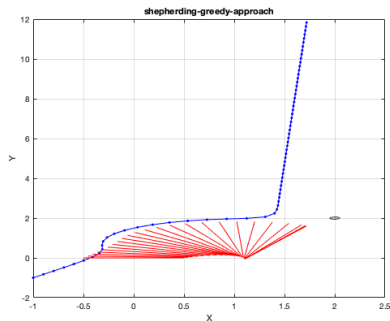


Figure: Shepherdng Greedy Approach

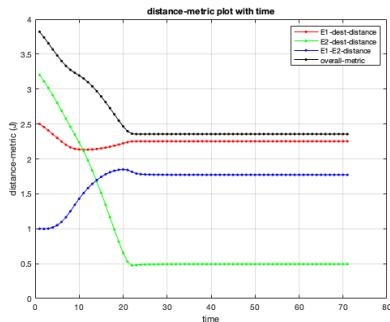


Figure: Plot of cost function and its components with time

Problem Flowgraph

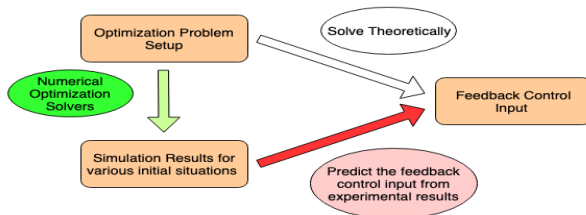


Figure: Problem Flowgraph

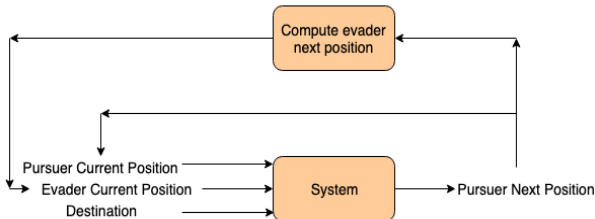


Figure: System Input-Output

Initial Condition

IPP : $(-1,-1)$, Z : $(-2,-2)$

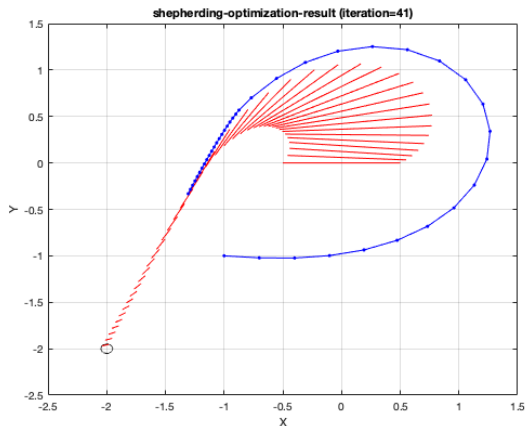


Figure: pursuer-evader trajectories

Initial Condition

IPP : (0,-2), Z : (2,-2)

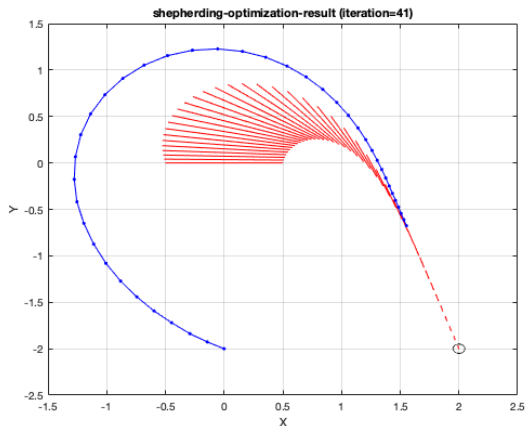


Figure: pursuer-evader trajectories

Initial Condition

IPP : $(-2,0)$, Z : $(-2,2)$

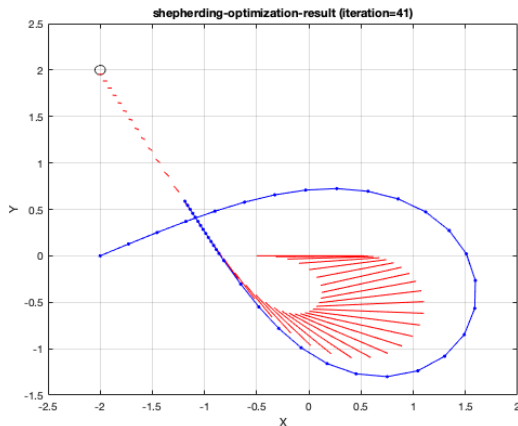


Figure: pursuer-evader trajectories

Initial Condition

IPP : $(-1,-1)$, Z : $(1,0.4)$

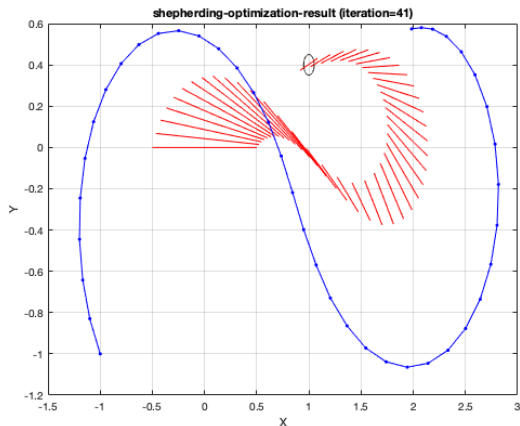


Figure: pursuer-evader trajectories

Initial Condition

IPP : $(-2,0)$, Z : $(2,2)$

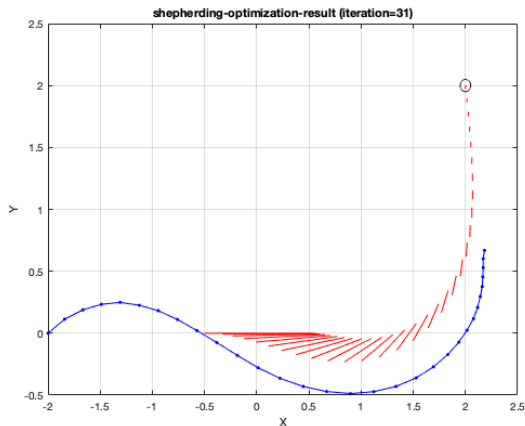


Figure: pursuer-evader trajectories

- Based on the above results, we can think of the two evader-system as a mass-dipole system
- The strength of the dipole can be modeled as magnitude of separation between the two evaders $= \|d(e_1, e_2, t)\|_2$
- The direction of the dipole can be taken as parallel or perpendicular to the evader-separation vector $= \hat{d}(e_1, e_2, t)$



Figure: dipole demonstration

- More intuition about the feedback input can be concluded by analyzing the evader trajectory in terms of dipole linear motion and alignment motion (rotation)

- The pursuer's tendency is to bring itself as well as the evaders in a straight line (approx.) with the destination

How to find or predict the final collapsing line based on initial conditions ? Can it be done ? How can this collapsing line be helpful ?

Triangle ΔPZE_c collapses nearly to a single line

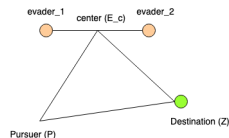


Figure: PZEc Triangle

- **Far Approximation case** : If the pursuer starts very far from the dipole, effect on the dipole in the beginning is negligible
- Significant (noticeable) movement of the evaders happens only when the pursuer is close

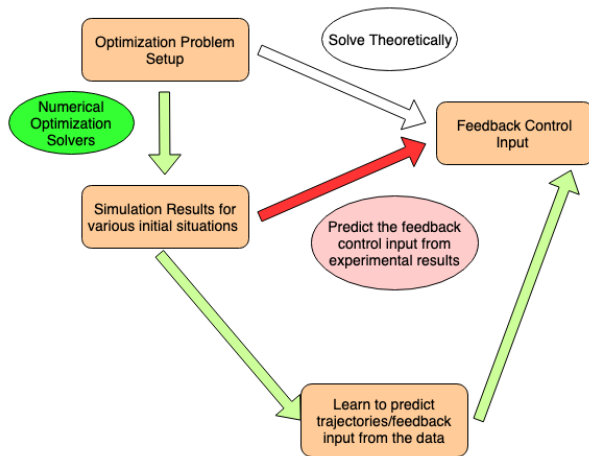


Figure: Problem Flowgraph

Model Architecture

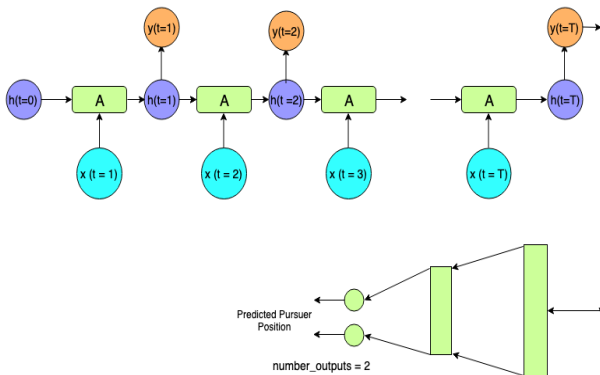


Figure: Model Architecture

RNN Formulation

- RNN state : $h_t = f_w(W_{hh}h_{t-1} + W_{xh}x_t)$, f_w is a non-linear activation function
- RNN output : $y_t = W_{hy}h_t$

- Input features : $x_t = [p_x(t), p_y(t), e_x^1(t), e_y^1(t), e_x^2(t), e_y^2(t), z_x, z_y, \dot{p}_x(t), \dot{p}_y(t), \dot{e}_x^1(t), \dot{e}_y^1(t), \dot{e}_x^2(t), \dot{e}_y^2(t)]$
- mean squared loss = $\frac{1}{M} \sum_{i=1}^M \|Y_g(i) - f_{\theta}(i)\|_2$

Intermediate conclusion on this approach

- The model is able to learn the following details from the data :
 - the smoothness of the trajectories
 - the fact that the pursuer approaches the evaders
 - Approx. the range of the velocity magnitude
- The model fails to learn :
 - the fact that the pursuer has to drive the evaders towards the destination
- Possible Reasons for failure :
 - Insufficient data
 - Model design and complexity (though based on my experience, the model is complex enough to learn these features)

- ① Try to formulate a feedback control law for the pursuer based on the observations or develop an iterative solution to achieve the same
- ② Generate more simulation results to improve the trajectory accuracy
- ③ Find out how to improve the learning model in order to put more emphasis on the destination point
- ④ Things to ponder about :
 - ① Dipole Alignment and Linear Motion
 - ② Triangle collapse
 - ③ Estimating the end-collapse line based on initial conditions

- 1 I would like to thank Prof. Debraj for constantly pushing me and giving the necessary directions
- 2 I would also like to thank Aditya Choudhary for all the help throughout this project

Thank you all for coming :)